VI. CLAIMS

I claim:

1.) A computing machine, comprising:

(a) a collection of computing elements called Effectors,

(b) a machine architecture which determines how Effectors behave and determines how information is transmitted from one Effector to another Effector.

2.) The machine of claim 1 wherein a subset of said Effectors, called Input Effectors, receive their information from one or more selected from the following: an external environment, a distinct Effector machine, a Static program, or a Meta program.

3.) The machine of claim 2 wherein in said Static program is implemented with an Effector machine.

4.) The machine of claim 2 wherein in said Meta program is implemented with an Effector machine.

5.) The machine of claim 2 wherein an input interpreter is used to help design one or more selected from the following: said Effector machine, said Static program, or said Meta program.

6.) The machine of claim 5 wherein in said input interpreter is implemented with an Effector machine.

7.) The machine of claim 1 wherein said machine architecture is designed using cyclic graph evolution.

8.) The machine of claim 1 wherein a subset of said Effectors, called Effectors, have their firing activity translated by an output interpreter.

9.) The machine of claim 8 wherein said output interpreter is implemented with an Effector machine.

10.) The machine of claim 1 wherein said machine is a dynamic machine.

11.) The machine of claim 10 wherein a Meta program changes over

time one or more of the following properties of said Effectors and said machine architecture: threshold, refractory period, pulse amplitude, pulse width, or transmission time.

12.) The machine of claim 1 wherein error tolerance is used to design hardware of said machine architecture.

13.) The machine of claim 12 wherein said error tolerance is used to build said hardware with transistors operating subthreshold.

14.) The machine of claim 1 wherein said machine architecture is designed using cyclic graph evolution.

15.) The machine of claim 14 wherein modules of Effectors are used to facilitate the crossover of two Effector machines.

16.) The machine of claim 14 wherein one or more of the following properties change for said machine or for Effectors in said machine during said cyclic graph evolution: number of modules per machine number of effectors per module, refractory period, threshold, number of connections, amplitude, pulse width, and conduction time.

17.) A computing method, comprising:

(a) providing a collection of computing elements called Effectors,

(b) providing a machine architecture which determines how Effectors behave and determines how information is transmitted from one Effector to another Effector.

18.) The method of claim 17 wherein a subset of said Effectors, called Input Effectors, receive their information from one or more selected from the following: an external environment, a distinct Effector machine, a Static program, or a Meta program.

19.) The method of claim 18 wherein in said Static program is implemented with an Effector machine.

20.) The method of claim 18 wherein in said Meta program is implemented with an Effector machine.

21.) The method of claim 18 wherein an input interpreter is used

to help design one or more selected from the following: said Effector machine, said Static program, or said Meta program.

22.) The method of claim 21 wherein in said input interpreter is implemented with an Effector machine.

23.) The method of claim 17 wherein said machine architecture is designed using cyclic graph evolution.

24.) The method of claim 17 wherein a subset of said Effectors, called Output Effectors, have their firing activity translated by an output interpreter.

25.) The method of claim 24 wherein said output interpreter is implemented with an Effector machine.

26.) The method of claim 17 wherein said method is a dynamic machine.

27.) The method of claim 26 wherein a Meta program changes over time one or more of the following properties of said Effectors and said machine architecture: threshold, refractory period, pulse amplitude, pulse width, or transmission time.

28.) The method of claim 17 wherein error tolerance is used to design hardware of said machine architecture.

29.) The method of claim 28 wherein said error tolerance is used to build said hardware with transistors operating subthreshold.

30.) The method of claim 17 wherein said architecture is designed using cyclic graph evolution.

31.) The method of claim 30 wherein modules of Effectors are used to facilitate the crossover of two Effector machines.

32.) The method of claim 30 wherein one or more of the following properties change for said architecture or for Effectors in said architecture during said cyclic graph evolution: number of modules per machine number of effectors per module, refractory period, threshold, number of connections, amplitude, pulse width, and conduction time.

33.) The method of using circuits of transistors operating subthreshold to implement an Effector machine.

34.) The method of claim 33 wherein Cyclic Graph Evolution is used to design said circuits.